CHAPTER

9 Information Rights Management

As described in the previous chapter, there are several technologies that address the security of unstructured data, but once that data leaves your network, those security technologies lose their effectiveness because you no longer control the environments where the data has migrated. One common theme among those technologies is that the security controls are tied to something other than the data itself, such as network or computer perimeters. As discussed, there is only one technology that fully secures access to the data regardless of where it travels. The solution is to build the classification metadata, the access controls, and the information about which rights are allowed to individual users right in to the data itself. This solution is known as information rights management (IRM).

IRM is essentially a combination of encryption and access controls that are built into document creation and viewing software applications, so that encrypted content can be decrypted and viewed based on access rights. In the following sections of this chapter, we examine the history of rights management technologies that began with the digital entertainment industry and led to today's IRM solutions that apply similar controls to any unstructured data.

In the first part of this chapter, we start with the high level architecture of IRM, including the primary components of any IRM infrastructure and how they work when a user is connected to the network and when they are offline. We will also discuss why auditing and reporting is an important feature of IRM.

In the second part of this chapter, we will look at the classification of data and how that leads to protecting data based on its confidentiality. We will also consider how users are given access to IRM-protected data and how that leads to locking down that data so it can be distributed to allow authorized users to access it, and what they are allowed to do with that data based on their rights assignment. You'll see how unauthorized users, who have no rights, are unable to do anything with the data, even when those users were previously authorized and their rights were subsequently revoked.

Overview

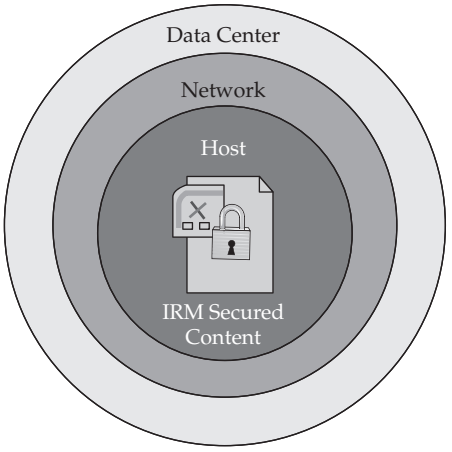

Figure 9-1 IRM shrinks the security perimeter of information to the content itself.

IRM shrinks the security perimeter to the information itself, as depicted in Figure 9-1. With IRM, you are not protecting the location where the information lives, nor the network it lives on. Instead, you are applying access control, encryption, and auditing to the information itself. That way, regardless of which disk the information resides on, which networks it travels across, or which database it may be resident in, IRM is able to provide a persistent level of security to the information wherever it goes.

IRM provides security protections not only for data at rest and data in transit, but also for data in use—which, as noted in Chapter 8, is hard to accomplish. IRM technologies are able to prevent things like data being copied to a clipboard and pasted into another application. IRM can allow authorized users to open content while also limiting their ability to edit that content or make printed copies of it. With this level of control for data in use also comes auditing of all access to the information, even after it has left the perimeters of your network. These controls are basically impossible to implement with any other technology.

With its fine-grained data-in-use features, the most valuable thing that IRM brings to the security landscape is the ability to control access to information, every time it is accessed, from any place it is copied to, and for every single copy, anywhere—along with the ability to revoke that access at any time. Imagine the scenario where your business has shared millions of e-mails, images, spreadsheets, documents, presentations, and so on with your business partners, customers, potential acquisitions, and employees (both current and long gone). Now imagine being able to revoke access to all that information and ensure that, as your business relationships and trusts change, you can maintain appropriate access to information even when it has long left the confines of your file servers, content management systems, and networks. The security of the data is persistent. Unlike nearly every other data security technology, the information is never given to the application or end user in an uncontrolled manner.

IRM technology extends the reach of information access control to well beyond places where you can typically deploy identity and access control technology. However, as with any technology, IRM has pros and cons, and this chapter will cover those. IRM is not a replacement for existing security solutions, but it is an excellent tool to complement them. IRM represents a powerful tool for reducing risk of data loss.

The Difference Between DRM and IRM

You may already be familiar with digital rights management (DRM)—the technologies used by copyright owners to protect music and movies. IRM is nearly the same thing; it can be considered a type of DRM. IRM is sometimes referred to as E-DRM, or enterprise digital

rights management. The distinction is that DRM has been developed mainly for use in the consumer space, while IRM focuses on the business problems of information security. DRM typically has a poor reputation with consumers, not only because of problems with over-restriction and excessive enforcement, but also because media companies who mandate its use have done a poor job of communicating any value of DRM to their customers. Thus, the IT world prefers the term IRM (or just rights management), to avoid negative associations with DRM.

The customers who buy music and movies from the entertainment industry feel strongly about being able to use, copy, and play their music and movies on any device they own. People are used to buying and owning vinyl records, cassette tapes, CDs, and DVDs—physical media that can be used in any supported player. Standards for CDs enable them to be played on any CD player, and the owner owned the content for the lifetime of the medium, which in theory is forever for a CD if handled carefully.

But does the owner of a CD really have a license to play the content for the lifetime of the CD? The answer to that is complicated, because the information owners (the entertainment companies) typically do not want consumers to have unlimited rights to play the content wherever they want. Entertainment companies have, on occasion, attempted to restrict playback ability even when the law and usage agreements do give consumers those rights. As a result, the owner of the CD is likely to believe they are allowed to do anything they want with that CD, whereas the entertainment company that produced the CD may not agree. Thus media rights are never completely black-and-white, and consequently are often disputed in courtrooms.

Most people believe that physical ownership of CDs can be transferred without restrictions—in effect, they are allowed to sell their CDs online or to their local music store. But how do the information owners view that (especially considering that the sales of used CDs can be considered as competition to the sales of new CDs)? In fact the entertainment companies generally do not agree with the consumers on this point.

The same debate exists for movies on DVD and digital media, and is even more heated because movies represent a more profitable product. Despite the fact that consumers are accustomed to being able to resell or give away their CDs without restriction, whether or not the record companies agree (and largely because not a lot of attention has been given to this type of rights transfer), those consumers don't understand why they can't do the same with digital media like movies. Ripping a movie is just as easy as ripping a CD, which is a very common way to move music from a purchased CD to a portable music player. Why is it okay to rip music but not movies?

From the perspective of the entertainment industry, the ease with which digital information can be duplicated and distributed has caused enormous financial losses because copyrighted entertainment content has been widely shared without any compensation to the industry. DRM has been used to recover and maintain that control. Corporations in the entertainment industry argue that they need to maintain control over their content to make money to pay artists, producers, studios, and others involved in the creation of the media.

With these opposing perspectives of consumers and corporations, the topic of DRM is controversial. The entertainment industry likes it because of its inherent ability to restrict use and redistribution, but consumers dislike it for the same reason—they want the same level of flexibility and ownership that came with physical formats.

Gone Too Far with DRM?

In 2005, Sony BMG made headlines when it was discovered that some of its music CDs secretly installed Trojan software on computers that forcibly implemented DRM. When unsuspecting consumers inserted a music CD into their computer, the Trojan compromised the computer by installing a rootkit that blocked attempts to copy or play music outside of Sony's own music player. Widely considered to be poorly programmed, the rootkit was later found to provide a convenient hiding place for other viruses and malware.

Regardless of the ethics of using malware-like behavior to force a company's requirements onto consumers, which most would agree is not good for customer relations, DRM itself went from being unpopular to being symbolic of "Big Brother." By forcing DRM onto its customers in this way, Sony BMG exacerbated the already existing division between consumers and corporations, and this incident elevated DRM's already troubled reputation to that of global pariah.

A classic example of DRM in use is the inability to copy a song downloaded from an online music store. DRM may restrict the customer from copying or moving the song from one computer or music player to another computer or music player. The real goal is to prevent unscrupulous people from distributing the content to other people. But DRM often prevents users from making legitimate backups of CDs or DVDs, and it can prevent the viewing or listening of content on devices because of incompatibilities and differences between varying manufacturers and varieties of DRM. As a result, many legal battles have ensued between the media companies and consumer groups over the use of DRM technologies.

A key differentiator between DRM and IRM is the relationship between the content producer and the consumer (see Figure 9-2):

- DRM controls access to published entertainment content such as music, films, and e-books. The relationship between the author of the content and the consumer is one-to-many. This means that a single entity creates the content, and many people purchase and use it. DRM is intended to ensure that a person who purchases particular content on a particular device can't transfer or distribute a copy of that content to any other device, other than as authorized by the owner, without financial compensation to the content owner. In other words, one owner, many consumers.
- IRM controls access to information such as medical data, financial records, engineering research, marketing plans, and sensitive e-mail communication—typically within the umbrella of a single organization. The relationship between the author of the information and the consumer is more complex and is considered many-to-many. For instance, a VP of finance who is the trusted authority to create company financial information might be working alongside a trusted group of executives who collaboratively work on a set of documents that contain private financial data. The VP of finance and his or her trusted colleagues might be allowed

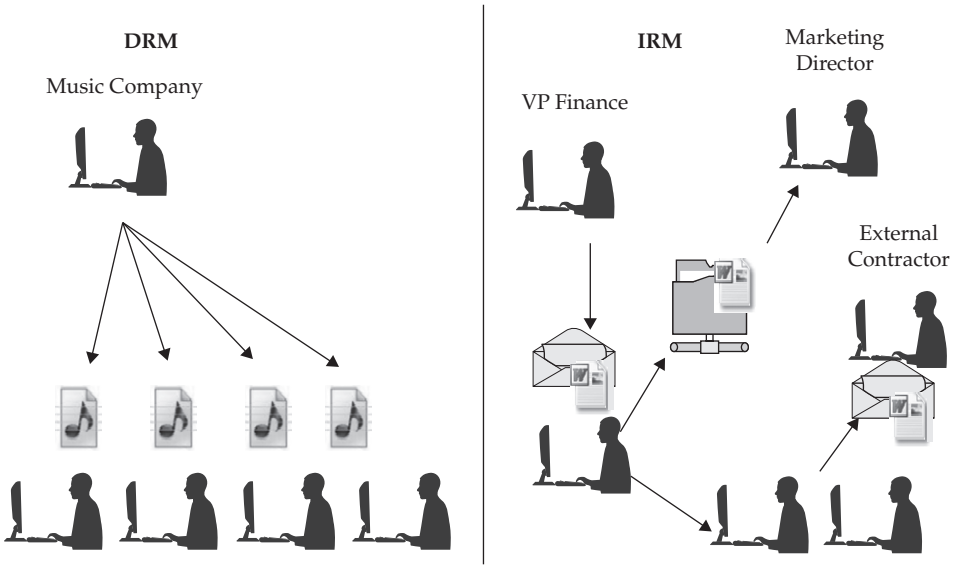

Figure 9-2 DRM and IRM user relationships

to open, edit, and print those documents. Yet a different set of people, such as the company sales team, may only be granted the right to open and view the contents of those documents, without being allowed to modify them. And yet another set of people, such as external contractors, might be allowed to access the location where those documents are stored, but they are not authorized to open the documents at all. In this example, the access controls of the storage system on which these documents reside will not be sufficient to provide the needed restrictions on what the sales team and external contractors can do.

The different access requirements of different sets of information consumers drive the designs of the technologies and their deployments. This is why IRM is more complex than DRM. DRM is all-or-nothing—either you have access to the data, or you don't. IRM controls what you can and can't do with data. A goal of DRM may not be a goal of IRM, and vice versa. People in general are frustrated with DRM because the main goal is to control and limit access to information, and end-user impact is not always a primary consideration. By contrast, IRM is something people want—to ensure that critical information about them or the company they work for is kept under control. DRM is typically aimed at controlling one file for access by one user, but in a business environment, information is shared with many authorized people and the access rules change often, thus IRM is a good choice.

What's in a Name? EDRM, ERM, RMS, IRM

Given that DRM has a bad reputation, and that security practitioners who like the benefits of rights management don't want to get embroiled in controversy, we would do well to avoid altogether the terms "digital rights management" and "DRM" and their

problematic associations. Thus, instead of using these terms, we need a new name. Several choices have been proposed:

- **Enterprise digital rights management (EDRM)** This is an attempt to associate the DRM concept with the goals and objectives of enterprise environments. It is considered a weak choice because the name doesn't really avoid the negative connotations associated with DRM.
- **Enterprise rights management (ERM)** This is similar to the preceding term but without "digital" included. The acronym ERM is already commonly used to represent the terms enterprise risk management and environmental resources management, so this is also a weak choice.
- **Rights Management Services (RMS)** This actually refers to Active Directory Rights Management Services (AD RMS), Microsoft's IRM technology. Microsoft uses the acronym AD RMS to describe the product that delivers IRM functionality. This is a brand name, therefore, this term should not be used to describe rights management in the broader sense.
- **Information rights management (IRM)** As you are already aware, in this book we've chosen to use this term, which is generally descriptive and provides a conceptual separation from the problematic and unpopular DRM technologies, even if it is just in name, and even then only by one word.

It's also generally acceptable to shorten the name to just "rights management," which is the common denominator among all the names, even though that term doesn't exactly make clear what "rights" are being "managed."

Evolution from Encryption to IRM

Many people think of IRM as a new technology, but it's actually been around for over a decade. Here's a brief history.

The first practical computerized use of encryption to protect unstructured content was PGP, developed by Phil Zimmermann to secure messages and files from prying eyes. This moved strong cryptography from the province of a few government agencies into the hands of millions of individuals around the world.

Other encryption solutions soon followed, some commercial, and others open source. Files wrapped in encryption could be shared electronically over the Internet and only those with the right cryptographic certificates could decrypt and access the data. However, file encryption solutions only allow the secure storage and transit of information—once the file is decrypted, the control is lost. It could be stored in unsecure file formats and easily retransmitted further without its protective encrypted wrapper. Furthermore, anybody with the right key could access the data. Keys could be lost, stolen, or published, and they need to be shared, so they didn't really constitute a bulletproof solution. In fact, key management became one of the hottest topics in encryption, for exactly that reason.

In order to preserve the protection of files regardless of what actions people might perform, the next main advance in unstructured content security was Digital Rights Management (DRM) developed and used by the entertainment industries as discussed in the previous section. In addition to its unpopularity, DRM proved eminently hackable.

All modern implementations of DRM have been compromised, resulting in an array of software tools available to end users to bypass built-in content controls.

IRM was the next and current generation. Securing not only the delivery of data, but also creating mechanisms of persistent control, IRM solved the problem of key management as well as restricting functionality. Consider, as an example of business requirements, a hypothetical organization that needs an IRM solution.

A company has an engineering department of 500 people, of which 300 are managers and product specialists who are allowed to create and edit classified information about the products they design and build. The remaining 200 people are only allowed to open and view that information. In addition, there are 100 people external to the company, at different partner companies, who also need access to some of that information. After 6 months, due to a change in the market, several external partners are cut, and 10% of the internal employees are laid off. A year later, the company acquires a smaller company of which 50 new people now require access to the information. In this time, some of the engineering department members have been promoted to a manager or specialist and as such they have gained a need for access to the information changes, allowing them to create and edit documents and emails. Over this hypothetical two years, thousands of documents, spreadsheets, images, emails and other such unstructured content is created for many different projects, all containing a variety of sensitive information.

How are you going to ensure that every copy of every document can be opened and used in a consistent manner? How do you ensure that terminated employees can't just walk away with all the companies' crown jewels and take it, en mass, to the competition? What about partner relationships that have changed? Where you've already sent thousands of sensitive documents? This is the challenge to be solved by IRM. The above-defined functional requirements are all needed in this complex scenario, which is representative of the complexity of a typical business case. Encryption, file collaboration services, and access controls alone won't be able to solve all the aspects of the business problem.

IRM Technology Details

Different vendors have different solutions, each with variations in architecture, but to be considered an IRM technology, an IRM solution should have the following characteristics: content encryption, identity-based rights definitions based on authentication, persistent access controls for data, granular rights, offline capability, and auditing and reporting. Let's consider each of these in more detail, beginning with an overview of the architecture of an IRM implementation and ending with a note about file formats.

What Constitutes an IRM Technology?

With over 20 technology vendors selling some kind of IRM solution, and many more general security vendors offering similar, non-IRM capabilities for protecting information in files and e-mails, how do we classify a solution as IRM? Off-the-shelf products range in capabilities from at-rest and in-transit encryption to fully configurable document functionality restriction. In the middle are many commercial solutions. So when is IRM really IRM?

The best definition of a full IRM solution is a document protection technology that supports the most commonly used business document formats, works when a user is connected

to the network as well as offline, allows revocation of access to content no matter where it resides, and includes all of the following criteria which combine controls for confidentiality, access control, and functionality.

1. Employs a client/server architecture that provides centralized management of rights (as opposed to a built-in protection that cannot be changed remotely).
2. Uses a format that includes the document content as well as metadata containing security rules that are used by document viewing and editing applications to control access.
3. Provides confidentiality for protected information in unstructured files through the use of encryption.
4. Leverages an identity from an enterprise directory to authenticate a user to the information and apply the access control. Embedding this access control into the content is not sufficient. Because IRM technologies are aimed at the enterprise, they should support industry-standard authentication schemes, work with identity federation technologies, and be able to connect to commonly used user authentication directories.
5. Applies a rights model, based on an enterprise classification system, that controls access to ensure only certain subjects can decrypt and access the information and define what they can do within the document. This should include a combination of at least the following basic in-use controls:
   - Create a new document with IRM protection based on a pre-defined classification
   - Open and view a document only; unable to edit or copy information within it
   - Edit and save changes into the protected format
   - Print to a trusted print device
   - Forward a document or message and reply to a protected message
   - Provide a basic level of screen-capture protection
6. Generates reports of access to content.

Architecture

The metadata that accompanies an IRM-protected document comes from both the IRM server and the IRM client performing the protection. The server provides a rights model and classification system to define the relationships between content and users. This is an important aspect of IRM (and it is one of the functional differences that distinguish IRM from DRM technologies—DRM creates a right for a user to access a single piece of information and restricts the user from interacting with the content and sharing it with others).

IRM, however, abstracts much of the data onto the server. This is because IRM must allow for secured information to be shared with many diverse parties, and the rights given to those parties can change often. IRM also must allow a wide range of collaborative scenarios, involving varying levels of rights, to occur without changing the document to which those rights are applied. If IRM were to embed the rights and identity data directly into the files, like DRM does, this would limit scalability to support the wider range of use cases required

in an enterprise document-rich environment. IRM also needs to integrate with complex business environments with overlapping requirements to define rights to information.

The system has to incorporate the enterprise identity infrastructure and also be accessible by certain parts of the business and end users. Unlike DRM, which is usually intended to be hidden from the end user, IRM wants to inform the user about the rights they have—and create mechanisms in the rights system that allow users to request rights changes from the content owner. When users have not been granted access to IRM-secured information, there needs to be a way to redirect users to the right business process to request access.

IRM solutions have a common set of basic architecture building blocks that are designed to facilitate access to the data for authorized users while blocking unauthorized users from performing various tasks with the data.

Clients and Servers

IRM technologies protect information that will travel far beyond the perimeters of the typical enterprise network. Therefore, IRM technologies use a classic client and server architecture. Figure 9-3 shows an example of how the client and server communicate to control access to protected content.

The IRM server stores information about user rights, cryptographic keys, auditing data, and classifications. The IRM server is accessible from the public Internet, so users can open documents no matter where they are (as long as they are on the Internet). Such access ensures that traveling employees can still gain access to content without the need to be physically connected to the corporate network.

Unstructured content is secured with the IRM server. This process typically happens in one of two ways: it is secured as part of an automated process when the content is downloaded from an application, a content management repository, or a file share; or end users secure

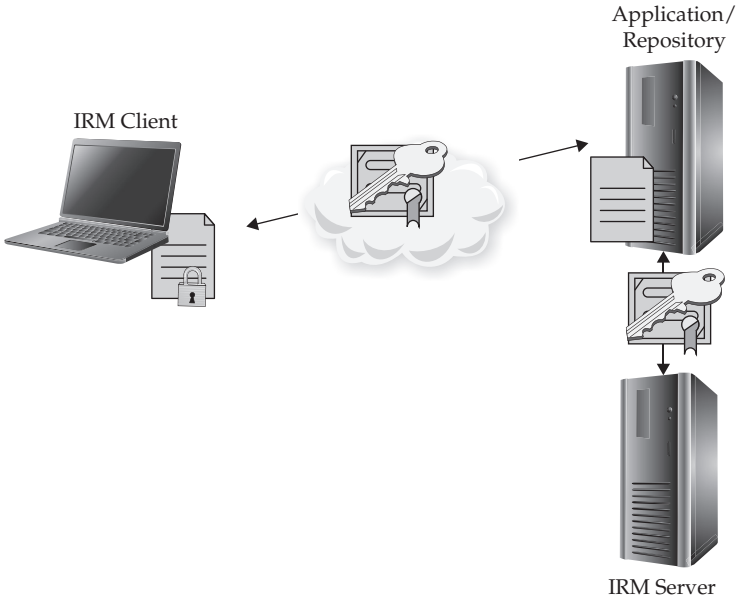

Figure 9-3 Client/server communication flow

content by making a manual decision using the client software user interface. Once the document is secured, it can be distributed anywhere with the confidence that only authorized users can open and use it according to the content owner's rules.

When a user attempts to open a secure file, a piece of client software is needed on their local computer to perform the decryption and enforce the access rules. Depending on the vendor, this may require an additional download or it may already be built in as part of the operating system or document publishing application. The client software reads the IRM server information from the file and communicates with the IRM server, passing authentication information and other document attributes. Depending on the information passed, the user may then be granted access to the document. If so, the information required to open the content is securely sent back to the client, where it may be cached for later use when the user is offline.

IRM Secured Content Format

IRM is all about protecting unstructured data files, as defined in the preceding chapter. E-mails, spreadsheets, documents, images, and HTML pages are examples of such unstructured data files. Each of these file types has its own specific format, but they all need to be protected in the same way. Most IRM vendors do not own the document format specifications for the content they are protecting. There is also no commonly accepted standard for how vendors should create IRM content, so it's up to each vendor to define its own IRM file specification. These specifications typically take the form of a container in which the information being protected is kept in encrypted form, in addition to metadata that describes what IRM classification applies to the protected content, along with the URL for the IRM service and other IRM-specific data. In some cases, the file extension for the document also changes. Figure 9-4 shows how encrypted data is commonly embedded in encrypted form in an IRM file that also contains rights data along with digital signature information to validate transactions.

Microsoft owns the format specification for Microsoft Office documents, and Adobe owns the specification for PDF documents. As such, those manufacturers didn't need to create new formats to secure documents; they chose instead to modify their own formats to include the IRM functionality, and design their software clients to enforce it. They still use encryption along with a section of new metadata containing the certificates, licenses, and other information needed to describe and control the content, as depicted in Figure 9-4. Other vendors have a similar approach. Once the source content has been encrypted and any rights metadata added, then the whole file is digitally signed to ensure it is tamperproof.

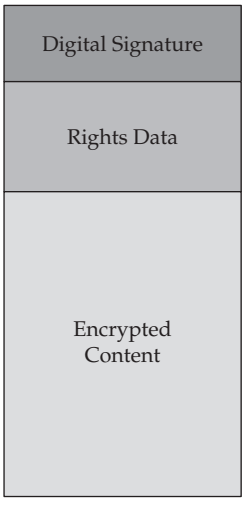

Figure 9-4 A typical IRM stack

Signing of the whole file is performed because while the source content is encrypted, the metadata needs to remain readable in order for the software to process it. We don't want that metadata, which contains classification information, to be changed so an attacker could modify their own permissions. To prevent such tampering, the entire document is signed so that any change in the bits causes the IRM client software to deny any access to the content, when it can no longer be trusted.

As an example, consider the protection of a JPEG file using Oracle IRM. The original JPEG file, which has a .JPG extension, is

Figure 9-5 The file header from a JPEG file protected with Oracle IRM

encrypted and then stored in a file with an .SJPG (Sealed JPEG) extension. At the top of the file is stored a variety of cleartext data that describes the classification of the file, the location of the IRM server, and similar information. Preceding all of this is a cleartext statement that informs anyone who opens the file in something other than the Oracle IRM Desktop client that they must have the software installed and configured correctly to access the file. Figure 9-5 shows a sample of such a file.

Cryptography

Cryptography is an important tool for protecting the confidentiality of data. For this reason, IRM relies on cryptography, which is a core component of IRM solutions.

Content Encryption The first step in any IRM technology is to encrypt the content of the unstructured data you want to protect. The encryption used to secure the content is almost exclusively symmetric-key encryption, because it is fast, and decryption to access the content usually needs to be done quickly. Symmetric-key cryptographic algorithms, in which the same key is used for the encryption and decryption of the data, are designed to be fast. As mentioned earlier in this chapter, this presents a problem if encryption is all you are doing, because anyone who has (or cracks) the key can easily decrypt the content. As with standard key-management approaches, asymmetric (public and private) key pairs are typically used to then encrypt the symmetric key itself. Asymmetric cryptographic techniques are not used to encrypt the actual unstructured content, because they are much slower. Nobody wants to wait while opening or saving their document.

Signing As previously described, IRM file formats contain freely readable metadata, which allows the IRM software to extract the IRM server location, classification of the data, and other content-specific information. This has to remain unencrypted; otherwise, the client software has no way to know where to get the cryptographic keys needed to decrypt content. Because this data resides in cleartext in the file, the whole file is signed to ensure someone cannot spoof the IRM server URL or other data. With a signature, any changes made to the

file would result in the IRM client software not being able to verify the signature, in which case it would refuse to open the tampered content. Another benefit of signing is the confidence that the content was created from a trusted source, and as such is a legitimate IRM-protected document.

Communication Encryption IRM clients need to talk to IRM servers to send authentication information and in return receive rights information. Due to a lack of standards, the actual protocols used are proprietary to each vendor. However, they do send their proprietary IRM communication over standard protocols for securing the connections. Nearly all IRM solutions use Secure Sockets Layer/Transport Layer Security (SSL/TLS) between client and server. This allows for IRM deployments to leverage existing PKI infrastructure for client and server communication. Most IRM servers are implemented inside web application servers. This means that setting up an SSL certificate to secure the client to server communication is relatively easy, and in fact is often already in place out of the box.

IRM deployments can be configured to communicate using regular, unsecured HTTP. This is highly unadvisable because the communications are going to transfer the cryptographic and rights information from the server to the client, and if this data is compromised, it puts the entire IRM deployment at risk.

Offline Rights Encryption After a client has received rights to access content, those rights can be cached so that the same content can be opened without a valid network route back to the IRM server. This is an important aspect of an IRM solution, because even in today's "always connected" world, being able to access secured content without communicating to the server allows for scalability, ease of use, and redundancy. Offline access is covered in more detail later in the chapter.

When rights are cached for offline use, they need to be protected. These rights contain the cryptographic information used to decrypt the secured content, so it is very important this information is protected from abuse. Once again, cryptography is used to solve the problem—an encrypted secure container is maintained by the IRM client software to store these rights. The encryption of the store is also typically keyed to the host computer in some fashion to ensure that the IRM-protected files can't simply be transferred to another, unauthorized computer to allow content to be accessed without the server authenticating the client.

Identities

Once a document has been secured with an IRM technology, access to it can be limited to a range of specified users, or *identities*. An identity might be a user in Active Directory, an X.509 certificate, or a user who's been authenticated by an online service like a social networking site. IRM uses these identities to enforce who can access what. This is different from just storing a password in a document. The identity and authentication process does not rely on the content of the document itself, and as such it allows for IRM solutions to revoke access to secured content without having to change the content. This is a great advance over plain file encryption, which allows access forever to those who have the key or can crack it.

This authentication process is a key benefit of IRM—it authenticates people with the same set of credentials they already use to access their corporate network, e-mail, and computer. Imagine a scenario where instead of using IRM, you use traditional file encryption to secure five documents that you share with a group of ten users. If the security solution places a separate password on each document, then you end up with five different passwords that all ten people

need to know. This would become even more complicated as the number of documents increased. So, what most people do instead is use just one password for all documents. That results in a single password protecting your most valuable information. If someone gets hold of the password, they can access the content regardless of whether they're supposed to. And what happens when you want to revoke access to any of the ten authorized people?

Identity Management (IDM) systems not only extend the functionality and ease of use of IRM, they also extend the reach of the IDM system to information beyond the traditional network perimeters. In a traditional network-resource environment, identity and access control solutions apply to information only as it resides within a network, device, or application to which security controls are applied. IRM extends the reach of those systems to anywhere the information may travel, inside or outside the network.

Another feature of IRM is the ability to revoke access to information. When an employee leaves a company, their corporate account is typically disabled or deleted. If that company protects its documents and e-mails with IRM, the employee automatically loses access to that information no matter where it has been transferred to. This solves the age-old problem of an ex-employee taking intellectual property from the company when they leave and giving it to a competitor.

The integration with identity management systems does stop at the corporate directory. IRM can be integrated with advanced risk-based authentication solutions that make access decisions based on numerous factors such as IP address, GPS location, or patch level of the host. Such integration can allow access to sensitive documents and e-mails only when the user is using a device within the borders of a particular office, area, or country, or in other known locations.

How IRM Authentication Works At a high level, here is a generic flow describing how IRM authenticates users:

1. A user attempts to access an IRM-protected file by opening an e-mail or double-clicking a protected file.
2. Before the content can be opened, the IRM client requests credentials from the user. This may require interaction with the user, or it may be an automated process leveraging an existing authenticated session and therefore achieving a state of single sign-on.
3. Credentials, combined with metadata about the content being accessed, are passed by the IRM client to the IRM server for validation over a secure network connection.
4. The IRM server (usually a web application server) authenticates the credentials against a connected identity store.
5. If the authentication is successful, the IRM server then determines whether the user actually has rights to open the content. Alternatively, the IRM server may rely on another identity management and access control technology to make this determination.
6. If the authorization is successful, a set of rights is passed to the user that enables the decryption of the content and rendering of it to the user based on the restrictions defined in those rights.

Access Feedback Loop Authentication and authorization are two different things. The former establishes who you are, and the latter defines what you can do. It's possible for a user to successfully authenticate but fail to authorize for a specific function. This may come

into play in collaborative environments where people need to know how to request access to information. The denial of access can also be leveraged to inform users about policy, the reasons for denial, and the exception request process. IRM works as an awareness-raising service as well as an access control tool. IRM works alongside identity management systems to redirect users back to self-service functionality, where a user's access request may go through a workflow that includes approval from the information owner.

Figure 9-6 depicts how the IRM client takes user credentials and passes them securely to the IRM server, which in turn authenticates the user against whatever identity platform the server has been configured to work with. The IRM server may be configured to trust claims from the user as a result of federation with external identity platforms the IRM server also trusts. After the user's identity has been authenticated, the IRM server checks to see whether the user is authorized to access the content. This may be another external call but often is a result of internal IRM server logic. If the user does indeed have rights to the information, the IRM server ships them a set of rights, which includes the cryptographic keys that allow the decryption of content. This is how the key-management problem is solved.

Rights

Document encryption and access controls are important components of IRM, but alone they do not constitute an IRM solution. This needs to be said, because some products may be marketed as "IRM solutions" when they really don't have full IRM functionality. Among the other features required to qualify a product as IRM is the rights model. This is the complex, granular set of rules that defines which identities are allowed to access what information (the secured content), and what they can actually do once they've been given access. It is this rights model that provides a logical mapping from your enterprise security policy to the

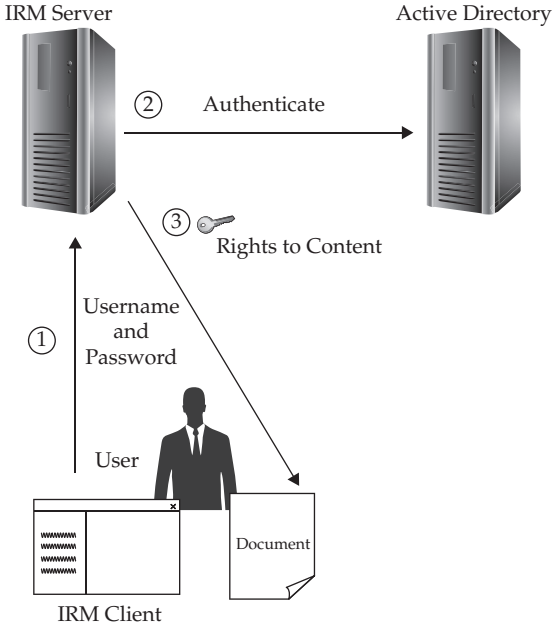

Figure 9-6 IRM client and server communication with a directory store

IRM technology. By contrast, file encryption only protects individual documents to be shared with a few other people, and it only scales to a small number of people. When you scale the challenge of document protection up to a group size of hundreds of thousands of users, you run into a management problem.

The IRM rights model allows you to create classifications, separate out groups of people in the business environment inside and outside the network perimeter, and associate the protected information with your security policy. By centralizing this policy management, you can retain control over access to thousands of documents in a manageable manner.

Rights classification is typically defined by three main components:

- The content being protected
- Rights defining access to the content
- Identities (users, groups) that are assigned rights

A data model then defines the relationship among these components. For example, you might create a classification in the rights model called Top Secret Finance Data. Within it you might add two groups from Active Directory, Finance Department and Executive Board, as shown in Figure 9-7. The Finance Department is given rights to open content, create content, edit content, and print, while the Executive Board is only given rights to open content. All of this information is stored in the IRM server, and certain aspects of the data are pushed to the IRM clients and into IRM-protected content.

Once a piece of content has been secured, a rights classification has been applied, and a user can access the content with their enterprise identity, an important aspect of IRM comes into play—persistent control. Instead of simply decrypting the information for the user to open in whatever application they choose, the IRM client software manages the entire process of decryption, rendering, and user interaction. After a successful authentication, the IRM server distributes to the end user a set of rights that dictates how they can interact with

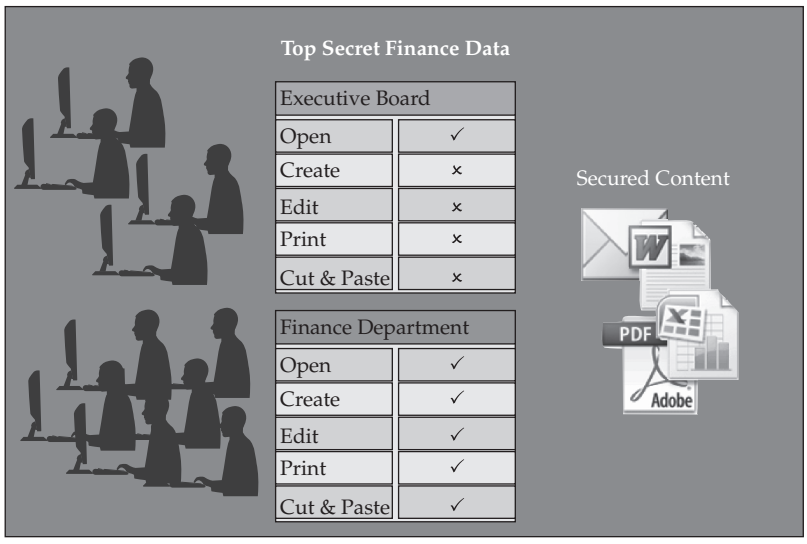

Figure 9-7 A simple example of an IRM rights model controlling functionality

the content. These rights will expose a varying level of functionality regarding the ability to print, copy, or edit the information.

First, the content needs to be securely handed off to the application that will control the interaction between the user and the content. This process typically goes like this:

1. The user successfully authenticates to the content, and a set of rights is made available to the user.
2. The IRM client accesses the source content by using the decryption keys for the content, which are contained in the set of rights. The content is decrypted into a secure location.
3. The application associated with the content (such as a document viewer) starts, and the IRM client either dictates to the rendering application what features should be made available or takes over the rendering directly (depending on the specific product). For example, the IRM client determines whether the print buttons in the interface and the print options in the menus should be available to the user.
4. Once the rendering application has been secured, it is passed a reference to the secure location of the decrypted content, and it renders the content to the user.
5. As the user interacts with the content and the application, the IRM client consistently monitors for legitimate and illegitimate activity. For example, the IRM client determines based on the user's rights what happens if the user attempts to take a screenshot, or what happens if the user is allowed to print but attempts to print a Word document as an electronic PDF file instead, which is really a way of saving the document in a different format. Figure 9-8 shows an example of this restriction.

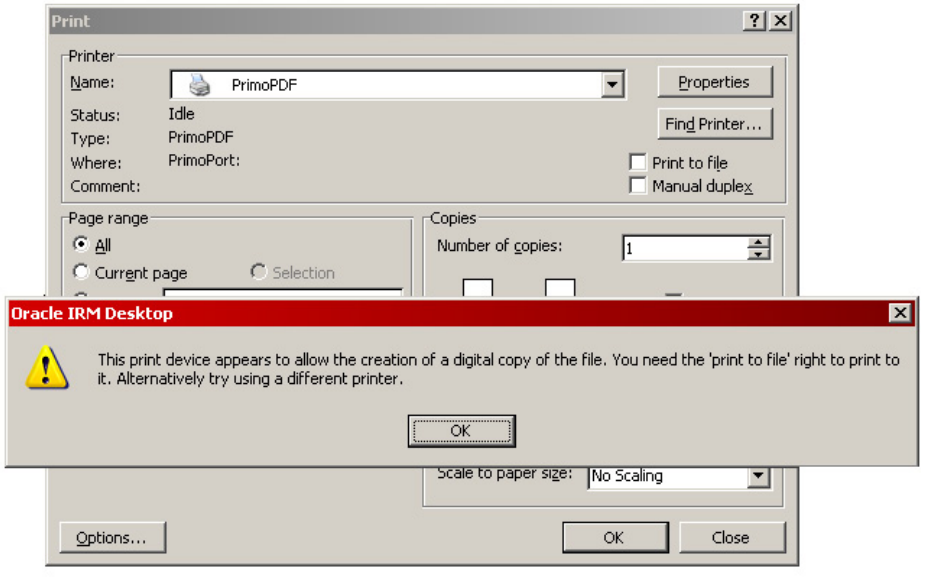

Figure 9-8 IRM prevents someone with print rights from creating a digital version via a PDF printer driver.

6. If the user is able to edit and save the document, the IRM client protects all the artifacts of this activity as well. It monitors and secures temporary files generated by the application, and encrypts any changes saved into the IRM secured file.
7. Finally, when the user closes the application, the IRM client ensures a clean exit—removing decrypted content from the secure location, destroying any cryptographic keys that were in use, and shutting down IRM-related processes.

IRM rights control individual actions that an authenticated user can take within the protected document. The rights described next are common to IRM solutions.

Create and Protect Some IRM technologies allow you to dictate whether a user has the right to create a new secured document, as opposed to opening an existing one. Being able to control this aspect of an IRM technology has a powerful implication. You can control not only who can consume protected content, but also who can create sensitive information.

To illustrate this principle, consider a scenario where you are using IRM to secure company e-mail communication. You may, for example, have a classification set up that allows all company executives to read e-mail announcements from the CEO. Let's say this classification is called Executive-Announcements. In this scenario, you only want a few select people, say the CEO, the CEO's administrators, and a few trusted executives, to be able to create and secure e-mail against this classification. For this particular classification, you don't want anybody else to have the ability to send executive announcements. You can then use the Executive-Announcements classification to restrict that ability.

The Create right differs from the Save right; instead of allowing someone to open, edit, and save changes to existing content, the Create right controls the authority to create new information in a particular classification and protect it.

Open and View Controlling access to open content and to view it is the most fundamental capability of an IRM solution. The Open right allows an authenticated user to open and view the protected content. In general, any user who is going to work with and access IRM-protected information needs the right to open and view it.

Edit and Save The right to open and view content only gives users the ability to read the information. The Edit right gives them much more capability by comparison. However, this capability is constrained by the underlying document management software. For PDF documents opened in a viewer, the Edit right may not have any effect at all, but for Word documents, the Edit right will enable the user to make changes to the content just as if IRM were not applied. There may also be differing levels of Edit rights. For example, one level may allow the user to copy data to the clipboard and other nonprotected applications, while another level may prohibit this action.

Most people would agree that if you have the right to edit a document, you should also have the right to save it—otherwise, your edits would be lost. Thus, edit and save rights are usually delivered at the same time. However, there may be unusual situations where you want to allow a document to be edited but not saved. For instance, you may want to allow a user to interact with a spreadsheet, enter data, and do calculations, but you may not wish these changes to be saved back to the official source document—much like a template that behaves like a calculator. In that case, you can prohibit the Save and Save As commands but allow the Copy to Clipboard right, so that data can be copied out of the template.

Alternatively, you could allow the user to use the Save As command, but prevent them from using the Save command—thus they could make their own personal copy based on something they changed.

Print Despite the growth and migration of information into digital form, and periodic promises about the "paperless office," people continue to want to print content. Printing is also one of the vectors by which content is easily lost, and it's one of the more difficult ones to control. For this reason, controlling the ability to print is perhaps the most popular feature of IRM solutions. Print features in IRM solutions can be implemented in various ways. Some IRM solutions will detect the conversion to other digital formats through the print driver, such as PDF print devices, and prevent this activity, while still allowing printing to actual physical printers. Preventing any printing at all is also an option. The right to print can be subject to further controls, such as mandatory automatic insertion of dynamic watermarks (like "Confidential" or "Do not distribute") into the content, thereby adding unique identifiers into the printed copy.

Whether or not to grant rights to print can be a contentious issue. Users want to be able to print documents freely, but organizations are often concerned about the proliferation of uncontrolled copies that granting such rights may produce. The decision to enable this right usually represents a balance between the usability of information by users and the need to protect its confidentiality. Some IRM solutions have functionality that enables administrators to dictate how many times a document can be printed, but in practice, limiting the number of printed copies is really not an effective approach. If you give a user the ability to print once, a simple trip to a Xerox machine defeats any IRM controls for future printing. Ultimately, the decision should be straightforward: if the content is too sensitive, don't let people print it.

Forward and Reply Because many IRM solutions are able to protect e-mail in addition to documents, there are usually e-mail-specific rights as well. Once again, the implementation of these rights differs from vendor to vendor. Some solutions have the ability to control whether someone can reply to a secure e-mail, while others have rights for allowing or preventing the forwarding of e-mails. Some have more complicated rights restrictions, such as controlling the ability to modify, copy, or export the content of the e-mail you are replying to when replying to an IRM-secured e-mail.

Screen Capture One of the first questions many people ask when they first hear about IRM technology is, "All that content control is just fine, but how are you going to stop somebody from simply printing the screen to get around copy and print restrictions?"

The answer is, IRM products can control screen capture in addition to built-in document functions. This helps avoid the side-channel approach to getting around security by working outside the document software framework.

This restriction can also be used to prevent sharing of the screen during meetings, presentations, online web meetings, and instant message chats.

Auditing and Reporting

The IRM capability to report on document access activities is one of its most useful features. Even implementing IRM and giving everyone full access rights to all content, with no restrictions, can have significant value in just the reporting capability itself. An access report

can contain details of what was accessed, when it was accessed, and by whom, along with details such as the activity involved (Create, Open, Print, Save) or where the content was accessed from (IP address, location on disk). This information is not available with any other data security solution, even DLP (Data Loss Prevention, which as mentioned in Chapter 8 reports on access only, not usage). With IRM, you now have a total view into the use of your most sensitive data. Some solutions even record attempted access to IRM-protected content, so you can also see who is trying to get access even if they are denied. Offline access to content can also be recorded and then sent back to the IRM server when the client next accesses the server. This type of reporting is useful both for keeping internal records and when working with compliance requirements or legal cases.

What if Somebody Takes a Picture of the Screen?

Photography is another side-channel approach to avoiding IRM controls that springs to many people's minds when they hear about IRM. "IRM sounds great," they say, "but what if I just take a picture of the screen?"

While it's true that IRM can't prevent somebody from taking a picture of the screen, it certainly makes stealing or leaking documents much more difficult. Instead of simply attaching a document to an e-mail or copying it to a USB drive, a person has to work a lot harder to steal information from an IRM-protected document. Like in an old spy movie, they would have to take a picture of each page, one at a time.

Furthermore, an authorized individual would have to first open the document. That means a trusted insider is involved. Without IRM, anybody could steal your data, but with IRM, it can only be done through a trusted channel. And, you have an audit trail, so you can find out who opened the document at the time the pictures were taken.

Finally, if somebody has to resort to taking pictures to steal information, they are willfully breaking the law (especially if the document is clearly labeled confidential) and the activity is more actionable than with unprotected documents because of the audit trail and the evidence that IRM provides. You can point to the thief and say "you stole this information" because their actions were clearly overt, and they can't use the "I didn't know" defense.

The built-in reporting features of IRM solutions may even lead some organizations to avoid restricting access to their data, at least initially, so they can record access and collect valuable information about usage patterns. Organizations that implement IRM initially may want to limit printing, prevent people from editing, and lock down all access to information. However, just the fact that IRM-protected content informs users that the information has been secured with a technology that can control and track all use of information may be a good initial step to changing user behavior and reducing the risk of data loss. Knowing that the company gets reports on how many times each user accesses the confidential information has an immediate psychological effect on users that motivates them to treat IRM-secured content differently. People are less likely to forward an IRM-protected document and are more likely to think twice about whom they intend to send it to. It doesn't matter whether the recipient would be able to open the content or not; just the fact that IRM is protecting it causes users to be more cautious about how they handle it.

They become aware, for example, that if they have rights to edit and print an IRM-protected document, and choose to do so, someone, somewhere in the company will know they have done so. This awareness can lead to positive changes in user behavior.

Going Offline

These days, computer devices are usually connected to a network of some form—the Internet, wireless, cellular…even airplanes are providing network connectivity. It is rare for a device to be totally offline anymore. However, occasionally people need to work with unstructured content offline, albeit for only short periods. There are also situations where a user may be online but unable to reach the IRM server to validate rights. As such, it is important that an IRM technology allow for clients to handle the access of IRM-protected content without requiring a direct network connection to the IRM server.

When assigning a right to access content, the IRM technology usually allows the definition of some offline period. This time period dictates for how long the access to content remains while the user is offline. When the period expires, the IRM client requires the user to regain access to the IRM server to validate their continued access. Providing this grace period is optional—you may wish to give users no offline period at all, and force them (for certain content) to always communicate with the IRM server. Or, for less sensitive information, you might go the other way and allow infinite offline access—but you probably would not do this for highly confidential documents.

Allowing offline use, and choosing the allowed duration, should take into account that sometimes you'll want to revoke access (such as when an employee leaves the company). As long as the user is offline, they can continue to access the content until the next time they connect to the IRM server. You may choose to give certain individuals longer offline periods, either because they are more trusted or because of their job role. Conversely, you may choose to give contractors or temporary employees little or no offline duration.

Secure and usable offline use is not easy to implement, because of the need for the application to check rights with the IRM server. Leading IRM vendors have solved these challenges, as described below, which include:

- How to perform offline authentication and authorization to IRM content
- How to automate the caching to ensure usability
- How to protect the cached rights from tampering

Offline Authentication and Authorization

IRM products typically solve the problem of offline access by caching the rights that were granted by the server at some previous time. This typically consists of saving the rights data in a temporary location on the computer that was first used to attempt to access a protected document. When the user first opens the document, the IRM client contacts the server to check the user's credentials (authentication) and what they are allowed to do (authorization). Once it receives the response containing the authorization information, that information is saved somewhere (typically on the hard drive, in a temporary file). Subsequent attempts to open the same file while offline rely on the IRM software to check that cached information. Thus, the user must carry with them the original computer used to first open the document when they travel and need to go offline.

Automating Offline Rights Caching

Balancing usability with security can be especially important when using IRM solutions. When authorized users are traveling, or otherwise cut off from access, their IRM clients must ensure that any rights they need are cached so that they can access content they are legitimately allowed to. Some IRM technologies cache this information when you first access a piece of content (as described above) for a specific classification, or sometimes the offline rights may pertain only to a specific document. Alternatively, some products have the IRM server send all available rights to the IRM client, even when the user has not actually accessed any content yet.

Tamper-Proofing Cached Rights

The cached rights information that is saved in the computer's temporary location (such as in a temporary file as described above) is cryptographically protected so it can't be copied to another computer or exploited by somebody who has not previously been properly authenticated to the identity management system via the IRM server.

Unstructured Data Formats

Up to this point in our analysis of IRM technologies, we've covered the use of encryption to protect content, access controls to ensure the right person gets to access those encrypted files, and the rights which dictate how users can or cannot interact with the information. Now we consider the types of unstructured content IRM typically secures.

One of the challenges of IRM is that it relies on the client software (document editing programs like Microsoft Word and e-mail programs like Outlook) to follow the rules IRM provides. To provide controls over things such as opening, editing, clipboard access, saving, and printing, IRM needs to understand the format of the content being protected, and the applications that open and render the content need to support those controls. This means that the IRM client must integrate with those applications, on many disparate platforms, to ensure that persistent security is delivered for the duration of the time the content is accessed.

E-Mail, Office, and PDF

Each IRM product has a set of formats it supports. The most commonly protected are

- Microsoft Office (Word, Excel, and PowerPoint)
- PDF
- E-mail

The applications and formats supported by IRM products are well defined for the protection of Microsoft Office content. PDF documents are somewhat less supported, because the PDF format is an open standard and many applications can render PDF content. E-mail support varies as well. However, new releases and versions of these products, and sometimes even patches and hotfixes, may require a delay while the IRM vendors catch up to support changes in functionality. Choosing an IRM vendor will probably require analysis of which of your platforms and applications are supported.

Generic File Protection

IRM products support a variety of different formats and applications. Many of them also support generic file protection, which is for applications and formats that IRM technologies

don't currently support. It is an attempt to provide IRM functionality for a wider range of environments. IRM functionality, in theory, can be applied to any unstructured file and opened in any application. The advantage is support across a wider range of platforms for more formats. The downside is that some features of IRM may not work, because the IRM software is unable to completely integrate seamlessly with every third-party application to provide the granular rights control described in the previous section. However, a generic approach can support a broader range of formats to provide the primary value of IRM, which is extension of the enterprise identity infrastructure to unstructured information anywhere.

Getting Started with IRM

The following is a summary of the steps required to get started with IRM, from a freshly installed, blank service; to configuration of the rights model; to securing, distributing, and accessing content; and finally to the revocation of access.

Classification Creation

Before anything can be secured, a data classification scheme is required to sort the data into categories that can be used to apply rules. Some IRM solutions allow for the ad hoc securing of content, which requires no initial preparation of classification. But the best deployments are those that leverage a central policy or classification scheme that can be mapped to a corporate security policy. The creation of the classification is typically done by the security department, because it has the deepest knowledge of classification and handling practices, and the ongoing ownership and management of the documents within those categories can be owned by different business groups. Chapter 5, which covers security policies, includes a discussion of common data classification practices.

Figure 9-9 shows an example of how data classification may be implemented in a corporate environment in which there are four internal departments that want to use IRM to control access to their documents: Engineering, Finance, HR, and Sales. This company has five classifications for all information: Secret, Confidential, Proprietary, Controlled, and Public. These five classifications are sometimes referred to by their shorthand identifiers, L1 through L5. In this example, the Engineering department has three levels of sensitivity: Secret, Confidential, and Proprietary. The Finance department also has Secret and Confidential data. Sales produces some Confidential data and HR handles some Secret data. In this use case, assume that each department has differing requirements for rights based on how they intend their information to be used (such as: printing is allowed for HR but not for Engineering at the Secret level, and Confidential Sales and Finance documents allow copy and paste but Confidential Engineering documents do not).

This example depicts how the combinations of sensitivity levels and departmental requirements play out. The granularity of user rights within an IRM solution, coupled with a data classification scheme consisting of several levels of confidentiality and the complex needs of varying departments, results in permutations that are simplified by creating an IRM classification system. At the Secret level, three IRM classifications have resulted, called "L1 Secret Engineering Information," "L1 Secret Finance Information," and "L1 Secret HR Information." Likewise, Engineering, Finance, and Sales each have an L2 Confidential classification, and Engineering has one L3 Proprietary classification. There is also a

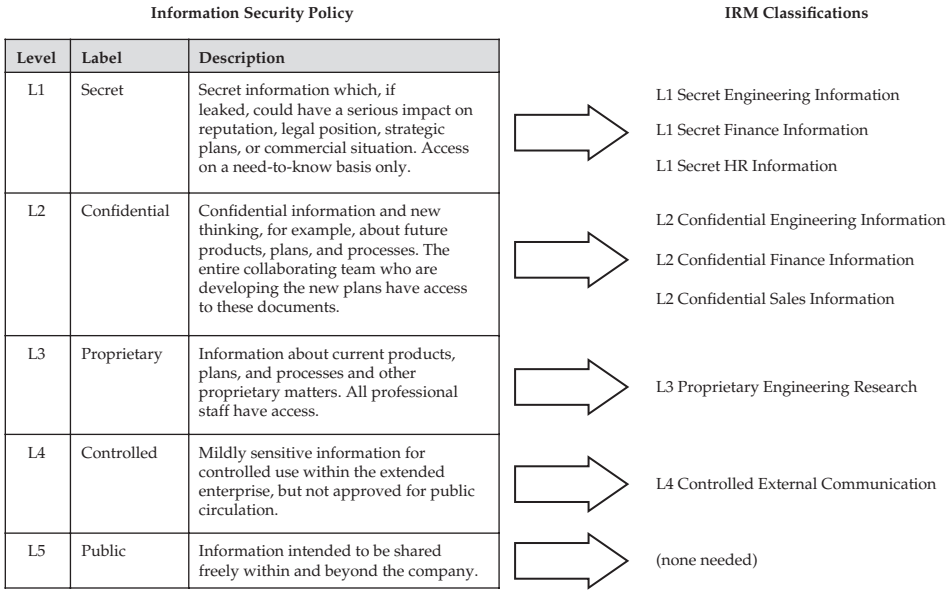

Figure 9-9 Data classification example

general category in this example called L4 Controlled External Communication, for documents that are classified as Controlled that are not tied to a particular department. And for documents classified as L5 Public, IRM is not needed, since these documents are intended to be open to all.

An IRM classification scheme translates complex use cases into simple designations that people can understand and use. People who create documents that use these IRM classifications will need to be trained on which ones to use, and when, but they won't have to worry about the underlying details. IRM classifications abstract the user rights to simple labels.

User Provisioning

You can't control access to content without distinguishing individual users so that they can be included in the access rights or excluded from opening documents. Thus, the IRM solution must be connected to a directory containing user accounts and must be able to authenticate incoming requests. It must also have a mechanism to authenticate external users, either federated with external user stores or by creating accounts in the corporate directory store. When creating accounts for external users, you must also communicate to them information about how to install the IRM client software and what their credentials are, if you haven't federated the IRM solution with their existing identity management solution. Provisioning may also involve the creation of groups that may be specific to the IRM solution.

Figure 9-10 demonstrates the necessity for user accounts to be in place before a user is allowed to access a document in an IRM implementation. In this example, the IRM server has received a request for document access from an IRM client, and is ready to authenticate the user who is requesting access.

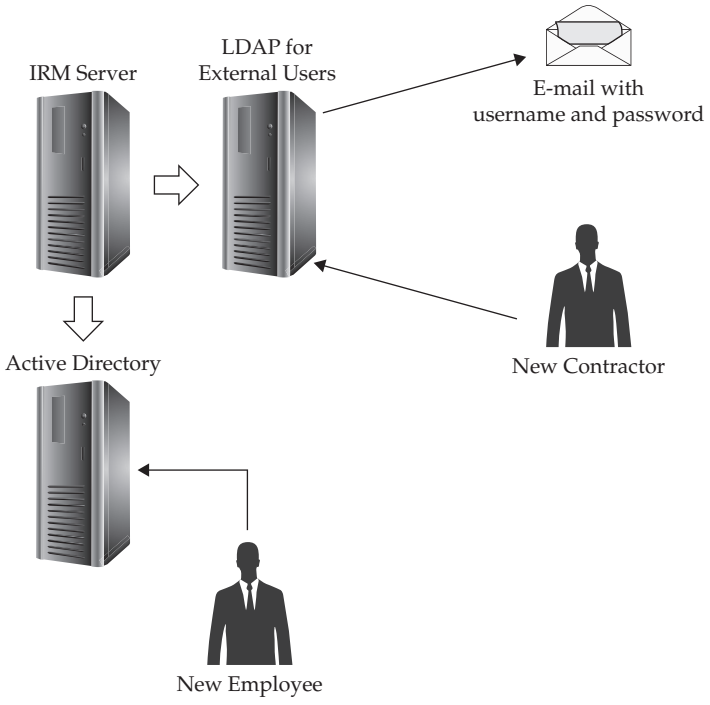

Figure 9-10 User authentication example

In this example, the IRM server has been configured to connect to both Active Directory (for employees) and an external LDAP directory (for contractors and other non-employees). If a new employee wants to open an IRM-protected document, a user account must be configured into Active Directory so the IRM server will be able to authenticate that person and determine what rights to grant. Likewise, if a new contractor needs access to protected documents, a user account must first be configured into the LDAP directory and the username and password need to be given to the contractor (in this case, via e-mail) so they can authenticate when they attempt to open the document. Of course, if the authentication attempt fails on both Active Directory and LDAP, all access to the document is denied.

Rights Assignment

With users available to the IRM server, you can now associate their accounts with the classification you've created and give them rights. Rights assignment can be done in a variety of ways by different members of the business. You can assign rights directly to individual users, but this creates administrative overhead and is not generally considered a best practice in any rights-assignment scenario. It is better to create or use existing groups and assign rights at this level, even if a group has only a single user account in it (see Figure 9-11). Then the management of groups in the directory has a direct reflection on which users have access to IRM-secured content.

The management of rights groups can vary among products. It may be defined by the security department or the IT department. Some IRM solutions allow for this management

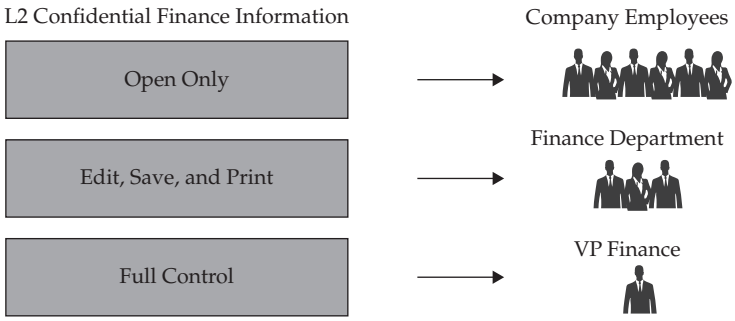

Figure 9-11 Rights assignment example

to be handed over to the owners of the information. The information owners may be in the best position to decide how they want their documents to be used.

Securing Content

The IRM solution is now ready to protect content. There are two typical approaches to doing this, as shown in Figure 9-12. One approach is to allow users to actively select information and choose the appropriate classification, and the other is to remove the decision from the user entirely and automate the protection. This is done through either integration with the application generating the content or by monitoring a file share and automatically securing files as they are stored within. DLP technologies are often integrated at this point to automate the creation of IRM control documents.

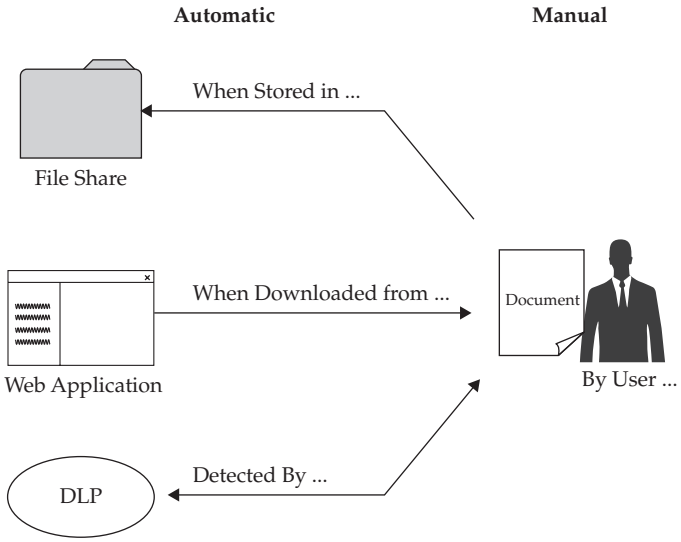

Figure 9-12 Automatic and manual content protection

Distributing Content

A key factor of IRM technologies is that, once documents are protected with IRM, it no longer matters how content is distributed. It can be e-mailed, posted on a website or file-sharing site, stored on a network file share, or even copied to a CD or USB drive (see Figure 9-13). There are numerous ways in which unstructured content can be distributed, and IRM works with all of them. Thus, the decision about how to distribute content to the end users can be based entirely on other business factors besides security.

Installing and Configuring the IRM Client

For large organizations with mature IT departments, it is usually possible to deploy the IRM client software automatically, at least to internal users. For some IRM technologies, the client software may already be in place—for example, in Adobe Reader or Microsoft Windows. However, you should consider the eventuality that at some point IRM-protected content will end up on the desk of someone who does not have the IRM client. They will then need to install the software and configure it so it can communicate with your IRM server. Most IRM solutions automate the client configuration, either by auto discovery or when a piece of IRM content is first opened. When this is done, the server settings are stored without user interaction.

Authentication

Now that a user has a piece of IRM content and the software required to open it, authentication takes place. This might be transparent if the IRM client supports single sign-on (SSO). For users who are external to the organization and for whom accounts have been precreated, there needs to be a mechanism of communicating their username and password beforehand. A common approach is to e-mail these details with a temporary password, which

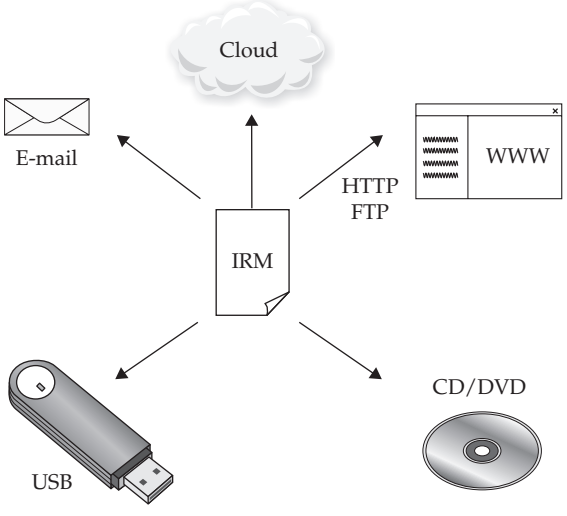

Figure 9-13 Examples of content distribution

Figure 9-14 User authentication example

the user is then prompted to change upon their first access to the IRM server (as depicted in Figure 9-10). This authentication phase may also take place offline and may differ from the online mechanism.

Figure 9-14 shows a login screen for an IRM system that is prompting the user for their username and password. This system allows the user to save their password for future attempts, in which case the software will pass the stored credentials back to the authentication server, and if they are still valid, the user will get access without having to re-type their password.

Authorization

Once authentication validates who the user is, the IRM solution must determine what rights they are authorized to have. This determination may be done with the built-in IRM rights model or it may be externalized to an existing application or technology that is already configured to understand who should get access to what.

Rights Retrieval and Storage

If the user is granted rights to the content, these are packaged up in a secure form and sent securely over the network to the IRM client to be interpreted. The rights package will also contain the cryptographic keys that allow the IRM client to decrypt the content.

If the rights are able to be cached and stored for offline use, they must be written to the host storage in a secure fashion and must be retrievable offline securely.

Content Access and Rights Invocation

At this stage, everything is in place to actually decrypt and access the IRM-secured content. The cryptographic keys need to be securely accessed and then used to decrypt the content. The rendering application then needs to be started and secured. The IRM client may need

Time	User	Function	Context	File	Status
9/30/2012 16:34	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	heat-source.docx	Success
9/30/2012 21:22	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	specifications.docx	Success
9/30/2012 21:51	mark.rhodes-ousley	Print	L2 Confidential Engineering Information	specifications.docx	Success
10/1/2012 18:58	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	heat-source.docx	Success
10/3/2012 10:34	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	specifications.docx	Success
10/3/2012 14:10	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	runtime.xlsx	Success
10/3/2012 18:58	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	results.pptx	Success
10/5/2012 17:46	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	design.vsd	Success
10/6/2012 21:22	mark.rhodes-ousley	Open	L2 Confidential Engineering Information	design.vsd	Success

Figure 9-15 Access auditing example

to hook into operating system functionality, such as the clipboard and screen-capture functionality. This all takes place before the decrypted content can be handed off to the application for rendering to the user.

Access Auditing and Reporting

Each time an IRM-secured document is opened, an audit record is generated, as shown in the example in Figure 9-15. These records may be generated on the server in real time as the request is made, or they may be cached offline and sent back to the server on a scheduled basis.

Rights Revocation

Finally, at some point the user will no longer require access to the sensitive data. At this point, their rights are revoked from the IRM server. This change in rights may happen immediately, if the client is online, or it may need to wait until the offline period expires, forcing the IRM client to communicate back to the server before the document can be opened, at which point the client discovers that the rights are no longer valid.

Summary

IRM technologies are a different, comprehensive approach to securing unstructured data. Unlike access control systems such as those built into file servers, or file encryption tools that require passwords and either grant all rights or none at all, IRM combines an entire layered security approach of access control, authentication, encryption, authorization, and auditing into a data-centric solution. By shrinking the access control perimeter from the network and storage to the content itself, IRM is able to enforce access and the security of documents and e-mails no matter where they reside.

The future of IRM technologies and their adoption may depend on the continued expansion of format support. Some vendors are evolving the platforms to a degree where they can support many of the basic IRM rights controls such as Open, Print, and Edit with a thin and simple application integration. This approach allows for the support of many formats, while giving up some of the finer-grained controls. Enterprises are also continuing to see increasing numbers and diversity of devices in the enterprise, including tablets and smartphone formats, which need to be supported in a comprehensive IRM deployment.

The security challenges of unstructured content are increasing along with the continued proliferation of unstructured data, and IRM is a good tool to deliver a persistent level of access control to information regardless of where it is and where it goes.

References

Becker, Eberhard, et al. *Digital Rights Management: Technological, Economic, Legal and Political Aspects*. Springer, 2004.

Harte, Lawrence. *Introduction to Digital Rights Management (DRM); Identifying, Tracking, Authorizing and Restricting Access to Digital Media*. Althos, 2006.

Smallwood, Robert, and Barclay Blair. *Safeguarding Critical E-Documents: Implementing a Program for Securing Confidential Information Assets*. Wiley, 2012.

Umeh, Jude. *The World Beyond Digital Rights Management*. British Informatics Society, 2008.

Zeng, Wenjun, Heather Yu, and Ching-Yung Lin. *Multimedia Security Technologies for Digital Rights Management*. Academic Press, 2006.